Homework 02 **Himen Hitesh Sidhpura**
COMP6651 **40091993**
Prof. Tiberiu Popa Due Date: 12/Apr/2019

1. Exercise 22.2-8

   Let x and y be the two end nodes of tree.
   To compute diameter of tree, we will use BFS algorithm with queue to find a longest path between two nodes.
   Step 1 : Run BFS algorithm from root node of tree to farthest nodes x.
   Step 2 : Now consider x as root and run BFS again to farthest node y and Consider diameter equal to 1 at root level.
   Step 3 : Enqueue node children in queue and increment diameter by 1 when.
   Step 4 : Once we reached y, we will get longest path in tree which is diameter.
   Since complexity of BFS is $O(n)$. Overall Complexity to computer diameter will $O(n)$, where n is number of nodes in tree.

2. Exercise 22.3-13

   Step 1 : Create a list of vertex in graph.
   Step 2 : Perform DFS for every vertex as a source in graph and mark them as visited in list.
   Step 3 : While performing DFS, If visited vertex encountered again. We can conclude that graph is not singly connected. If there is no encounter of visited vertex then we will perform Step 4.
   Step 4 : Repeat Step 2 for all vertex which are unvisited in list. Step 5 : It is singly connected.

   Hencem Overall Complexity of this algorithm is $O(V^2)$, here V is vertex.

3. Exercise 23.1-11

   Step 1 : Add a newly decrease edge in tree. Addition of this edge will make the cycle. This will make tree once again graph.
   Step 2 : To identify the cycle, we will use DFS. This DFS will give number of vertex in cycle.
   Step 3 :After identify cycle, we have to remove any edge along the cycle, which will gives us minimum spanning tree.
   Step 4 : We will remove a edge which has maximum weight along the cycle.
   Step 5 : After edge is removed, we will get a tree back having minimized weight.

   Hence, Overall Complexity will be $O(V + E)$ where V is number of vertex and E is number of edge

4. Exercise 26.3-4

   Let consider that there exist matching in G. Then each vertex of A will matched with R for any subset $A \subseteq L$. There will be no two vertices matched with R, since there exist a matching. Therefore, there exist only $|A|$ vertices in A.

   Now consider for all $A \subseteq L$, $|A| \leq N(A)$. Run Ford-Fulkerson on flow network. When augmenting path is found, flow in incremented by 1. This should happen for $|L|$ times. Suppose, it happen less then there will no path. Then there exist fewer L edge from L to R having flow 1.

   The flow is increasing each time by f_p by corollary 26.3 and f_p is integer which is proved by theorem 26.10. and particularly it is equal to 1 which implies that $|f| \geq |L|$. Since, it is clear that $|f| \leq |L|$.

By corollary 26.11, the cardinality of maximum Matching in G is equal to value of maximum flow f. Since, $|L| = |R|$, maximum matching is perfect matching.

References

[1] https://www.geeksforgeeks.org/diameter-n-ary-tree-using-bfs/

[2] https://www.geeksforgeeks.org/connectivity-in-a-directed-graph/

[3] https://www.geeksforgeeks.org/level-order-tree-traversal/

[4] https://www.geeksforgeeks.org/maximum-bipartite-matching/

[5] https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/

[6] https://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/

[7] https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols